EXHIBIT A

International Journal of Industrial Organization 11 (1993) 239–260. North-Holland

Complementary network externalities and technological adoption*

Jeffrey Church

Department of Economics, The University of Calgary, Calgary, Alberta T2N 1N4, Canada

Neil Gandal

School of Management, Boston University, Boston, MA 02215, USA and Department of Economics, Tel Aviv University, Tel Aviv, 69978, Israel

Final version received October 1992

We address the adoption of technology when there are network externalities and networks are characterized by complementary products produced by different firms. We show that when the software firms are Bertrand competitors, a hardware technology with lower software development costs is adopted for many parameter values for which it is socially optimal to adopt the other technology. The over-adoption is due to a discrepancy between the private and social benefit of having a larger network.

1. Introduction

The benefit received from the consumption of a particular good often depends on the aggregate number of consumers who elect to purchase compatible goods. This positive consumption, or network, externality can be direct or indirect. If it is the former, the utility of a consumer depends directly on the total number of subscribers to the same network. For instance, the value of access to a telephone network depends on the total number of consumers with similar access. Examples of goods where the network externality is indirect include many consumer electronic durables such as televisions, video cassette recorders, compact disc players, and personal computers. To be of value these durable or hardware goods require

Correspondence to: Jeffrey Church, Department of Economics, The University of Calgary, Calgary, Alberta T2N 1N4, Canada.

*We are especially grateful to Robert Masson for many helpful suggestions. We also wish to thank Richard Gilbert, Michael Katz, Jeff Perloff, Susanne Scotchmer, two anonymous referees, and seminar participants at Harvard University, GTE Labs, Boston College, University of Alberta, Boston University, University of Calgary, Canadian Economics Association Annual Meeting, Canadian Economic Theory Conference, and the Sixth World Congress of the Econometric Society. Church gratefully acknowledges the financial support provided by the Social Sciences and Humanities Research Council of Canada.

complementary software. The value of the hardware is enhanced as the variety of complementary software increases. Variety may in turn depend on the total number of consumers who purchase compatible hardware. The greater the sales of hardware, the greater the demand for software. This increases the profitability of supplying software. Entry by firms and the concomitant proliferation of software is the form of the network externality when it is indirect.

In this paper we formally model complementary network externalities. We show that with CES preferences over software, increasing returns to scale in the production of software, and free entry into the software industry the consumption benefit of *hardware* depends on the number of consumers purchasing the same hardware. Under the maintained assumption that software is not compatible between the two hardware technologies, we use this model to investigate how the interaction between the hardware and software firms affects the adoption prospects of two competing hardware technologies. Which technology prevails depends on the availability of software. If very little software is available even technically superior hardware will be relatively unattractive to consumers.

The compact disc digital audio system provides superior sound reproduction. Nevertheless, without the abundant provision of numerous recording artists on this new format by the record companies, compact discs would not be supplanting the phonograph. The success of compact discs can be contrasted with the failure of various quadraphonic audio technologies. The absence of a large variety of software halted the widespread adoption of these technologies.

Another example is personal computers. Despite its advanced capabilities and user-friendly interface, initial acceptance of the Macintosh computer was lukewarm. This was primarily attributable to the limited availability of software. All of Apple's subsequent generations, e.g. the Macintosh, were incompatible with the Apple II software base, providing a window of opportunity for the IBM personal computer to supplant the Apple II as the dominant platform.

IBM ensured that the introduction of the PC would be matched with the availability of plenty of software. This was attained by disclosure of the technical details of the computer and by adopting the MS-DOS operating system. The adoption of this relatively unsophisticated operating system simplified the task, and therefore the cost, of developing software.

Previous work in the literature has typically examined the implications of *single product networks* and the associated *direct network externality.* In a series of papers Farrell and Saloner (1985, 1986a,b) explore the demand-side coordination problems associated with network externalities. In order to fully realize network benefits consumers must correctly anticipate the adoption choices of other consumers. For example, a socially superior outcome may

have occurred if only one of the VHS or Beta video cassette formats had been adopted from the outset.

Katz and Shapiro (1985, 1986a,b) are concerned with the behaviour of oligopolistic producers in a single product network. They note that if network externalities exist there are obvious benefits if all consumers are on the same network or a compatible network. The common theme of their work is an examination of the social and private incentives to attain compatibility, i.e. standardization. In (1986a) they consider the case where there are two incompatible technologies and investigate whether the market, by adopting only one of the competing technologies, establishes a de facto standard. In their paper, one technology enjoys a cost advantage in the first period and a second technology enjoys a cost advantage in the second period. They show that when both technologies are supplied by a monopolist, the second technology is adopted for many parameter values for which it is socially optimal to adopt the first technology.

In this paper we examine networks which consist of both hardware and software products. The essence of such a network is described by three characteristics. First, we rule out vertical integration. Hardware firms cannot provide their own software. We do not explore this restriction, but note that it corresponds to the actual organization of most of the markets in which we are interested.[1]

Second, when faced with a choice between two networks, consumers typically evaluate each on the basis of the software expected to be available. We capture this aspect of preferences by adopting a symmetric CES utility function.[2] We have thus implicitly assumed that both hardware technologies are capable of performing the same tasks or yielding the same services, provided the requisite software is available. Furthermore, this specification of preferences assumes that consumers value all available software products equally. Hence a consumer cares only about the number of software products provided, the price of software, and the price of hardware.

Third, the production of software typically involves set-up or development costs which are large relative to marginal production costs. Given free entry in the production of software, the fixed costs of software development determine the extent of the network externality.

Within this 'hardware–software' paradigm, we address the following two issues: (1) How does the explicit introduction of software affect the technology adopted by consumers? We are particularly interested in determining whether the behaviour of the software industry has efficiency implications

[1] Church and Ware (1989) provide an explanation for this distribution of functions across firms based on the incentive problems associated with private information innovators have about their human capital.

[2] The CES utility function was first used to model variety by Spence (1976) and Dixit and Stiglitz (1977).

with regard to the network adopted in the market. (2) Does the behaviour of firms in the software industry result in the adoption of a network technology which is not socially optimal?

We show that when software firms are Bertrand competitors, a hardware technology that has lower development costs is adopted for many parameter values for which it is socially optimal to adopt the other hardware.[3] We establish that there is a systematic bias: the technology which is the most attractive to consumers in the market and the technology which provides the greatest social benefit to consumers are not the same.

An exception to the focus on single product network externalities is Chou and Shy (1990), who also consider a model wherein the network externality arises due to an increase in the number of complementary products. In their model hardware technologies are supplied competitively, software firms are monopolistic competitors, and the preferences of consumers for hardware are heterogeneous. They show that both hardware platforms can exist in equilibrium and that for some parameter values this two system equilibrium is Pareto inferior to a standardization equilibrium.

In our model both the hardware and software markets are oligopolistic and consumers' preferences over hardware are homogeneous. Our focus is on behaviour in the software industry and whether or not the market standardizes on the right technology.

In section 2 we describe the preferences of a representative consumer. We discuss the production technology in section 3. Section 4 is a description of the model. In section 5 we derive the equilibrium in the software industry. Section 6 is an examination of the market pattern of adoption. We characterize the socially optimal adoption of technology and compare it with the market pattern in section 7. Section 8 provides concluding remarks.

2. Consumer preferences

In this section we specify the preferences of a representative consumer. We then derive the software demand functions of a consumer and the benefit or utility they receive from joining a network.

There are two types of goods in the economy: the numeraire good and network goods (each network good is comprised of a hardware technology and compatible software). The two competing networks and their respective hardware technologies are denoted A and B. Since the software is not compatible between the systems, a consumer that purchases a unit of hardware A must purchase software written for that system. The benefit received from the hardware depends on the number of software products available.

[3]Under Bertrand competition, software firms take into acount the effect that changes in their price have on the price index of software. Under monopolistic competition, software firms ignore this effect. See footnote 13.

The preferences of a consumer are therefore represented by the utility function

$$U(x_0, z, \kappa) = x_0 z^{\kappa}, \tag{1}$$

where x_0 is the numeraire good and z is a network benefit function. The variable κ equals 1 if the consumer purchases either hardware technology A or B; 0 otherwise. The network benefit function is

$$z(x_1, \ldots, x_N) = \left(\sum_{i=1}^{N} x_i^{1/\beta} \right)^{\beta}, \tag{2}$$

where $\beta > 1$, x_i is the amount of software good i consumed, and N is the number of software products.[4] The greater β, the stronger the preference of a consumer for variety.[5] We have assumed that hardware only facilitates the consumption of software and therefore it does not enter the benefit function explicitly. No consumer purchases more than one unit of hardware and we assume that each network has the same benefit function.

The representative consumer who purchases hardware for system h ($\kappa = 1$) will maximize (1) subject to the following budget constraint:

$$\sum_{i=1}^{N_h} \rho_i^h x_i + x_0 + p_h = y. \tag{3}$$

where ρ_i^h is the price of software variety i on network h, p_h is the price of a unit of hardware supplied by firm h, and N_h is the number of software products available for system h. Consumers are homogeneous and y is the (common) endowment of the numeraire good.

Each consumer has a choice of purchasing one of the two competing hardware systems and some of *each* software variety available for that system. A consumer may also elect not to purchase a network technology ($\kappa = 0$), in which case his indirect utility is simply y. The alternative chosen will be the one which provides the greatest benefit. Henceforth we assume that the benefit provided from either technology exceeds y and thus every consumer purchases one of the hardware technologies. The system purchased will be the one which provides the greatest consumption benefit or indirect utility. We now derive the indirect utility function of a consumer who purchases network technology h.

Since both the network benefit function and the utility function are homothetic, two-stage budgeting is appropriate [Deaton and Muellbauer (1980, p. 130)]. The first-stage problem in which the consumer determines the

[4]The restriction $\beta > 1$ insures that (1) is concave.

[5]As β rises, the elasticity of substitution between software products [$\sigma = \beta/(\beta - 1)$] falls.

optimal budget share of the numeraire and the network has the following specification:

$$\max_{x_0, z} U(x_0, z, 1) = x_0 z$$

$$\text{s.t. } q_h z + x_0 = y - p_h,$$

where

$$q_h(\rho_1^h, \rho_2^h, \ldots, \rho_N^h) = \left(\sum_{i=1}^{N_h} (\rho_i^h)^{-1/(\beta-1)} \right)^{1-\beta}$$

is the price index of software for technology h.[6] The first-stage solution entails the consumer spending $(y-p_h)/2$ on software and $(y-p_h)/2$ on the outside good, i.e. $x_0 = (y-p_h)/2$.

In the second stage, the consumer optimally allocates the share of software, $(y-p_h)/2$, among the available software products:

$$\max_{x_i} z(x_1, x_2, \ldots, x_{N_h}) = \left(\sum_{i=1}^{N_h} x_i^{1/\beta} \right)^{\beta}$$

$$\text{s.t. } \sum_{i=1}^{N_h} \rho_i^h x_i = \frac{y-p_h}{2}.$$

The solution to this problem consists of the following system of demand equations:[7]

$$x_i[\rho_i^h, p_h, q_h] = \frac{(y-p_h) q_h^{1/(\beta-1)}}{2(\rho_i^h)^{\beta/(\beta-1)}}, \quad \text{for all } i. \tag{4}$$

Substituting (4) into (2) gives the consumption benefit of a network:

$$z(q_h, p_h) = \frac{(y-p_h)}{2q_h}. \tag{2a}$$

Substituting $x_0 = (y-p_h)/2$ and (2a) into (1), the indirect utility of a consumer that purchases system h is

$$V(q_h, p_h) = \frac{(y-p_h)^2}{4q_h}. \tag{5}$$

[6]See section A.1 in the appendix for the derivation of this price index. It is the Lagrange multiplier on the dual of the second-stage problem.

[7]This derivation is found in the appendix, section A.1.

The system purchased by a consumer (A or B) will be the one for which (5) is larger.

The consumption benefit of a network depends on the price index of software (q_h) and the price of hardware (p_h). The latter term determines the total funds available for the purchase of software. The price index of software depends on the price of each software variety and the number of software products available for hardware system h. Hence this term reflects the extent of the network externality. The greater the number of software products and the lower their prices, the smaller is q_h and the greater is the benefit from joining the network.

A salient case in what follows is when the price of each brand of software is the same. If $\rho_i^h = \rho_j^h = \rho^h$ for all i and j, then

$$q_h = \frac{\rho^h}{N_h^{(\beta-1)}} \tag{6}$$

and the indirect utility of a consumer is

$$V(q_h, p_h) = \frac{(y-p_h)^2 N_h^{(\beta-1)}}{4\rho^h}. \tag{5a}$$

This representation highlights the importance of the price of software (ρ^h), the number of varieties of software (N_h), and the consumer's preference for variety (β).

3. Production technology

Typically the creation of software involves large development costs. These fixed costs are orders of magnitude larger than the marginal cost of production. We denote the fixed cost of developing a software product for hardware h by $F_h, h = \text{A}, \text{B}$. We assume that the fixed cost of developing software for the two technologies is different, but that this cost is the same for all products developed for a particular hardware. We restrict each software firm to the production of a single variety.

The source of the complementary network externality is the effect that an increase in hardware sales has on the demand and hence the profitability of supplying software. The extent of the network externality will depend on the number of new software varieties provided when sales of hardware increase. We assume that software firms have unlimited or free access to the technology required to produce software. The number of software firms is endogenously determined by a free-entry condition and it depends critically on the magnitude of the fixed costs of software development.

The actual production cost for a unit of software consists only of the cost of duplication and packaging. We assume that these marginal costs are constant and equal across all software products provided for either hardware technology. We denote the magnitude of the marginal cost of software production by s.[8]

There are only two hardware technologies, A and B. Each is provided by a monopolist. The constant marginal cost of producing hardware is denoted by $c_h, h = \mathrm{A, B}$.

4. The model

Fig. 1 displays the timing and structure of the model. At the beginning of the period hardware firms set their prices. Consumers then purchase the hardware system that gives them a higher expected surplus. In order to determine the benefit derived from the adoption of hardware system h, consumers must, from (5), have expectations about q_h. Hence they must anticipate the software supply response – the number of software products and software prices – to their adoption decision.[9] In equilibrium, consumers' expectations will be fulfilled. There are m consumers who play an adoption game. Since consumers are homogeneous there will be two Nash equilibria to the adoption game. All consumers will adopt one hardware or all consumers will adopt the other hardware.[10] We assume that the equilibrium to the adoption game is the Nash equilibrium which is Pareto superior for consumers.[11] Based on the hardware adoption decision of the consumers, software firms develop software and set their prices, then consumers purchase software. The game is solved by backwards induction.[12]

5. The software industry

We begin with the determination of the equilibrium software price and the equilibrium number of software firms. The profit of a representative software

[8]Since the elasticity of demand for a software product is constant, for an equilibrium to exist, $s > 0$.

[9]Computer consumers for example, are often informed about likely software developments through computer magazines and advertisements. NEXT computer, for example, took out full page advertisements in the *Wall Street Journal* to announce that Lotus and Adobe – two prominent software firms – would be providing software for its platform.

[10]If consumers were heterogeneous, both systems could exist in equilibrium.

[11]For a detailed exposition of demand side coordination problems, see Farrell and Saloner (1985, 1986a, b).

[12]Our results are robust to changes in timing. The most natural alternative would be to have software firms develop software before consumers make their hardware adoption decision. Provided hardware firms can commit to prices, the equilibrium will be the same. Indeed, in order to induce software firms to incur development costs in this alternative timing, hardware firms will have to find some mechanism to make price commitments. One such mechanism is licensing or second sourcing. See Farrell and Gallini (1988).

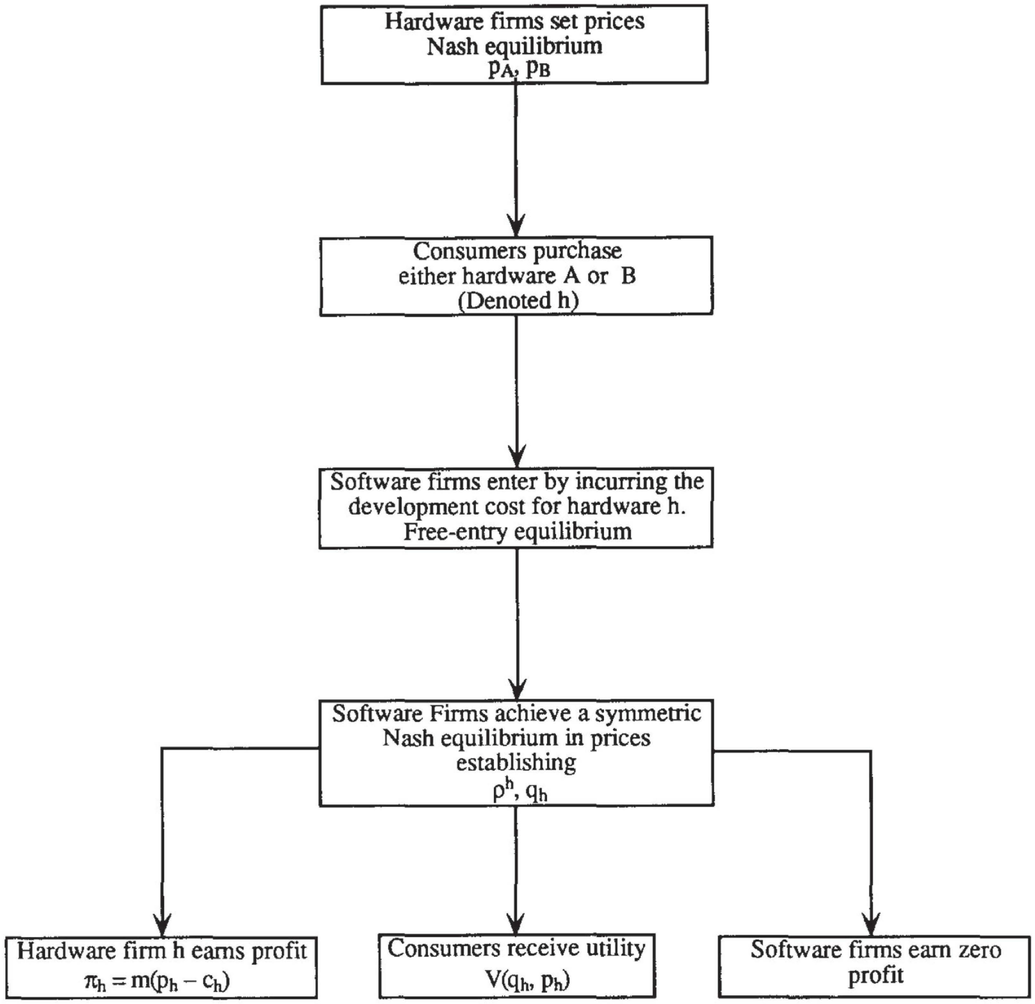

Fig. 1. Timing of the game.

firm i depends on the hardware system (h) adopted by the m consumers, the price it charges (ρ_i^h), the price charged by the hardware firm (p_h), and the price index of software (q_h), which summarizes all software prices:

$$\begin{aligned}\pi_i[\rho_i^h, p_h, q_h] &= mx_i[\rho_i^h, p_h, q_h](\rho_i^h - s) - F_h \\ &= \frac{m(y-p_h)(\rho_i^h - s)q_h^{1/(\beta-1)}}{2(\rho_i^h)^{\beta/(\beta-1)}} - F_h,\end{aligned} \tag{7}$$

where we have substituted (4), the demand of each consumer for a variety of software. Software firm i takes as given the price of hardware, the price of all other software varieties, and the number of software firms and chooses ρ_i^h to maximize profits. The best-response function for a representative software firm is implicitly defined by

$$\frac{\mathrm{d}\pi_i}{\mathrm{d}\rho_i^h}=\frac{\partial\pi_i}{\partial\rho_i^h}+\left(\frac{\partial\pi_i}{\partial q_h}\right)\left(\frac{\partial q_h}{\partial\rho_i^h}\right)=0.$$

Since competition in the software industry is Bertrand, when firms select their profit maximizing price, they take into account the indirect effect that changes in ρ_i^h have on $x_i[\rho_i^h, p_h, q_h]$ due to the dependence of q_h on ρ_i^h.[13] Assuming symmetry, we can state the following Lemma:[14]

Lemma 1. The symmetric equilibrium price of software is

$$\rho^h=\frac{(\beta N_h-1)s}{N_h-1}. \tag{8}$$

The proportion by which equilibrium prices are marked up over marginal cost is decreasing in N_h, the number of firms in the software industry.[15] If we substitute the equilibrium software price, (8), into (6) the magnitude of the network externality is

$$q_h=\frac{(\beta N_h-1)s}{N_h^{(\beta-1)}(N_h-1)}. \tag{9}$$

Substituting (8) back into (7), the equilibrium profit of a representative software firm under Bertrand competition is

$$\pi_i[p_h, N_h]=\frac{m(\beta-1)[y-p_h]}{2(\beta N_h-1)}-F_h. \tag{10}$$

A free-entry equilibrium requires that in equilibrium the number of software firms be such that each software firm earns zero profits.[16] The equilibrium

[13]Under conditions of monopolistic competition, firms ignore this dependency and treat q_h as exogenous: they act as if $\partial q/\partial\rho_i^h=0$. The monopolistically competitive price is βs. See Dixit and Stiglitz (1977). If the dependence of x on q is ignored, then firms are effectively acting as monopolists and it does not matter whether they choose quantity (as in Dixit and Stiglitz) or price.

[14]The derivation of Lemma 1 is in section A.2 of the appendix. We further restrict the parameter space so that $N_h>1$. Eq. (11) below shows that the necessary parameter restriction is $m(\beta-1)[y-c_h]>2\beta F_h$. This restriction does not depend on s. The parameter restriction implicit in our assumption that purchasing a network is optimal incorporates s.

[15]In contrast, the symmetric equilibrium price under monopolistic competition is invariant to the number of software firms in the industry.

[16]We ignore the integer problem. Our results would not qualitatively change if we restricted N_h to be an integer.

number of firms is found by setting (10) equal to zero and solving for N_h. The free-entry number of firms is[17]

$$N_h[p_h]=\frac{m(\beta-1)[y-p_h]}{2\beta F_h}+\frac{1}{\beta}. \tag{11}$$

We have the following proposition regarding the existence of a complementary network externality.

Proposition 1. Provided the price equilibrium in the software industry is symmetric, the welfare of a consumer on a network increases, ceteris paribus, as the aggregate number of consumers on that network increases.

Proof. From (11) it is clear that for any price of hardware, the number of software products provided increases as the size of the consumer cohort, m, increases. Since ρ^h is declining in N_h (see Lemma 1), the benefit of joining a network (5a) is increasing in the number of software products available for the network, N_h. Q.E.D.

For any hardware price, p_h, eqs. (8) and (11) determine the Nash equilibrium price of software and the free entry equilibrium number of software firms. Eq. (9) determines the extent of the network externality. In the next section we derive the equilibrium pricing strategies for the hardware firms and the implied pattern of adoption by consumers.

6. The market pattern of adoption

At the beginning of the period, the two hardware firms engage in price competition. Only the firm which can offer consumers the greatest surplus or benefit will have positive sales. When hardware firms lower their prices, they increase the benefit provided by their network by a direct and an indirect effect. From (5a), the indirect utility function, there is a direct effect from a lower price of hardware: it increases the expenditure available for both software and the numeraire good. However, there is also an indirect effect. An increase in expenditure by consumers for software increases the demand for software. In a free-entry equilibrium, more software firms will enter, thereby increasing the variety of software provided and the welfare of consumers. Moreover, an increase in the number of software firms will decrease the price of software, further increasing the welfare of consumers.

[17]The equilibrium number of firms in independent of s. This arises since gross equilibrium software profits are proportional to the price–cost margin, $(\rho^h-s)/\rho^h$. Since the equilibrium software price is proportional to s, the price–cost margin and hence the equilibrium number of firms is independent of s.

The lowest possible price a hardware firm will be willing to charge is its marginal cost: a lower price would result in negative profits. From (11) the equilibrium number of software firms increases as the price of hardware decreases. The *maximum* number of software firms that can be induced to provide software if network h is adopted is

$$\hat{N}_h = \frac{m(\beta-1)[y-c_h]}{2\beta F_h} + \frac{1}{\beta}. \tag{12}$$

It is worth emphasizing that the preceding expression is not the equilibrium number of software firms: $\hat{N}_h$ is the number of software firms (and hence software products) if hardware firm h prices at marginal cost and its technology is adopted in the market. We denote the magnitude of the network externality obtained from substituting (12) into (9), as $\hat{q}_h$.

The maximum benefit consumers receive from purchasing access to network h occurs when the price of hardware equals marginal cost and it is, using (5)

$$V(\hat{q}_h, c_h) = \frac{(y-c_h)^2}{4\hat{q}_h}.$$

If

$$\frac{(y-c_A)^2}{4\hat{q}_A} > \frac{(y-c_B)^2}{4\hat{q}_B}, \tag{13}$$

then the maximum surplus generated by network A exceeds that of network B. By (13) there is a $p_A > c_A$ such that

$$\frac{(y-p_A)^2}{4q_A} = \frac{(y-c_B)^2}{4\hat{q}_B}. \tag{14}$$

At this price (less epsilon) hardware A is adopted in the market and firm A makes positive profit. Solving (14) for p_A gives p_A^*, the equilibrium hardware price for firm A:

$$p_A^* = y - (y-c_B)\sqrt{\frac{q_A}{\hat{q}_B}}. \tag{15}$$

The equilibrium hardware price for firm B is $p_B^* = c_B$. At these prices no firm gains by deviating. Firm B would earn negative profits if it lowered its price and no profits if it raised its price. Firm A would earn zero profits if it raised its price and less profit if it lowered its price.

Network A will be adopted in the market if the surplus it provides consumers when hardware A is priced at marginal cost exceeds the surplus provided to consumers by network B when hardware B is priced at marginal cost. If technology A is adopted, then the equilibrium hardware price for technology A will be such that, in conjunction with the amount of software supplied at that price, consumers obtain the same benefit they would have received if technology B was adopted and priced at marginal cost.

Eq. (14) indicates that two technological aspects of a network are critical in the determination of its adoption prospects: the marginal cost of hardware production and the cost of software development. The expenditure available for software purchase depends on the price of hardware. The magnitude of the network externality depends on the number of software firms. The number of software firms is determined in large part by the fixed cost of software development. The trade-off between the fixed costs of software development and the production cost of hardware is captured in the following proposition.

Proposition 2. Technology A is adopted in the market if:

$$(F_A)^{(1-\beta)}[y-c_A](m[y-c_A]-2F_A)(m(\beta-1)[y-c_A]+2F_A)^{\beta-1}$$
$$>(F_B)^{(1-\beta)}[y-c_B](m[y-c_B]-2F_B)(m(\beta-1)[y-c_B]+2F_B)^{\beta-1}. \tag{16}$$

Proof. To derive (16), substitute (12) into (9) to find $\hat{q}_h$. Substitute $\hat{q}_h$ into (5) to find the maximum surplus associated with each hardware technology. Substitute this maximum for each technology into (13) and rearrange. Q.E.D.

Eq. (16) delineates the market adoption pattern as a function of the hardware firms' marginal costs, the software development costs, the preferences of consumers for variety, and the size of the consumer cohort. The role of the fixed costs of software development can be understood if we let $\beta=2$. Then technology A is adopted in the market if

$$[y-c_A]^3 > \frac{[y-c_B]^3 F_A}{F_B} - \frac{4F_A([y-c_B]F_B-[y-c_A]F_A)}{m^2}. \tag{17}$$

Suppose that technology A has lower costs of software development, i.e. $F_B > F_A$. Then for the market surplus to be the same for each technology, firm B must have lower marginal costs, i.e. $c_A > c_B$.

The network advantage provided by lower software development costs is manifested in two ways in (17). The coefficient F_A/F_B in the first term on the

right-hand side of (17) is less than one and it reflects the effects of the differing number of software varieties, holding the price of software constant.

When $F_B > F_A$ and $c_A > c_B$, the second term on the right-hand side of (17),

$$\frac{4F_A([y-c_B]F_B-[y-c_A]F_A)}{m^2},$$

is positive. This term reflects the lower price of software on the network of hardware A due to a greater number of software firms. When m^2/F_A is small, this term is relatively large since under these circumstances there will be relatively few software varieties supplied for *either* network. If there are only a few software firms, then an increase in the number of software firms has a large negative impact on the price of software and the equilibrium price of software if technology A is adopted will be significantly less than the equilibrium software price if technology B is adopted.[18]

7. The socially optimal adoption pattern and the efficiency of the market outcome

In this section we compare the socially optimal pattern of adoption with the market outcome. We consider two second-best social optima, corresponding to two different regulatory environments. In the first environment, the regulator or social planner can only mandate adoption of *hardware technologies*. In the second environment the regulatory authority has the power to regulate both the number of software firms, the price of software, and the price of hardware.[19]

The case when the regulator can only mandate adoption is straightforward. Suppose that the regulator asks each hardware firm to bid for the right to serve the market and gives the exclusive license to serve the market to the firm that provides consumers with the greatest benefit. The firm that wins the license will offer its technology at a price that just gives consumers the same benefit that they would receive if the other technology was priced at marginal cost. Since this is exactly how the market outcome was determined, the choice of the regulator and the technology adopted in the market will be the same. Consequently, the regulator cannot improve on the market outcome.[20]

The surplus provided by a technology characterized by complementary network externalities depends on both the conditions in the hardware and

[18]This is due to the convexity of the equilibrium software price with respect to the number of software firms in the industry. See (8).

[19]These two regulatory schemes are in the spirit of DeBrock and Masson (1985). No public subsidies are permitted in either scheme. The difference between the two is that in the second case, cross-subsidies between the hardware and the software industries are permitted.

[20]We thank an anonymous referee for suggesting this regulatory scheme and its implications.

the software market. In general the social benefit of a technology will be enhanced if the mandate of the regulator encompasses *both* the hardware and the software industry.[21]

We now derive the socially optimal pattern of adoption when the regulator can control the price of hardware, the price of software, and the number of firms in the software industry. In the optimal pattern of adoption, the technology which provides consumers with the greatest utility is provided. The problem of the regulator is to select the price of hardware, the price of software, and the number of software varieties to maximize the total benefit of the m consumers:[22]

$$mV(N_h^s, \rho_s^h, p_h^s) = \frac{m(y - p_h^s)^2 (N_h^s)^{(\beta - 1)}}{4\rho_s^h}, \tag{18}$$

subject to the break-even constraint

$$m(\rho_s^h - s)x_i[\rho_s^h, p_h^s, q_h^s]N_h^s + m(p_h^s - c_h) - N_h^s F_h = 0,$$

where p_h^s, ρ_s^h, and N_h^s are, respectively, the second-best socially optimal price of hardware, software, and number of software products. The gross revenues from the sale of software and hardware must cover the fixed costs of software development.

If hardware system h is adopted the utility of the consumer cohort is maximized by providing

$$N_h^s = \frac{m(\beta - 1)[y - c_h]}{\beta F_h}, \tag{19}$$

setting

$$\rho_s^h = s, \tag{20}$$

and

$$p_h^s = \frac{(\beta - 1)y + c}{\beta}. \tag{21}$$

[21]One industry where the scope of the regulator's powers includes the regulation of the software industry is the provision of video entertainment in residential homes in Canada. In this industry the access charge (price of hardware), the price of different bundles of channels (the price of software), and the number of bundles (variety of software) are all regulated. Moreover, the pricing of basic service apparently involves a subsidy from the cable companies to the software industry.

[22]As all software products are equally valued, the regulator will provide the greatest consumption benefit by selecting a common software price. Hence eq. (18) comes directly from (5a).

In the constrained social optimum solution the price of software equals marginal cost. The fixed costs of software are funded by elevating the price of hardware above marginal cost.

Proposition 3. If the regulator can control the price of hardware, the price of software, and the number of software varieties provided, the adoption of technology A is socially preferred to the adoption of technology B if

$$[y-c_A] > \left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)} [y-c_B]. \tag{22}$$

Proof. To derive (22) substitute (19), (20) and (21) into (18) for each hardware technology. Q.E.D.

The next proposition compares the adoption pattern in the market with the optimal choice of the regulator when the regulator has the power to set prices and control entry into software. We are interested in whether the technology that provides the greatest social benefit is adopted by the market.

Proposition 4. There is a set of parameter values for which the technology with the lower fixed cost of software development will be adopted in the market, but the regulator would choose the technology with the higher fixed cost of software development: The technology which has the lower fixed cost of software development is over-adopted.

Proof. See section A.3 in the appendix. Q.E.D.

The market pattern of adoption is biased towards technology A when $F_B > F_A$ because the price distortion (above marginal cost) in the software market is increasing in the fixed cost of software development. In the constrained social optimum, this distortion is reduced to zero for both technologies. This is done by raising the price of hardware and using the returns to subsidize software production. Consequently, in the market the advantage of a larger network is an increase in both the variety of software and a lower price of software. In the social optimum, an increase in software variety is the only benefit of a larger network.

The divergence in the benefits from a larger network between the market and the second-best social optimum can be illustrated by letting $\beta=2$. Recall that when $\beta=2$, the condition for technology A to be adopted in the market is

$$[y-c_A]^3 > \frac{[y-c_B]^3 F_A}{F_B} - \frac{4F_A([y-c_B]F_B - [y-c_A]F_A)}{m^2}. \tag{17}$$

From (22), when $\beta = 2$, the adoption of technology A is socially preferred if

$$[y-c_A]^3 > \frac{[y-c_B]^3 F_A}{F_B}. \tag{23}$$

Suppose that technology A has lower software development costs and that firm B has lower marginal costs of hardware production, such that the social surplus provided by the two technologies is the same, i.e. (23) holds with equality. The second term on the right-hand side of (17),

$$\frac{4F_A([y-c_B]F_B - [y-c_A]F_A)}{m^2},$$

is positive, since $c_A > c_B$ and $F_B > F_A$. The market surplus of A is greater than the market surplus of B, even though the social surplus is the same for each technology. Hence there are parameter values for which technology A is adopted in the market while technology B is socially optimal.

The inefficient adoption arises due to a price effect. The technology with the greater number of software varieties in the market will provide greater surplus due to both a greater variety of software and a lower price of software. The technology with the greater number of software varieties will provide greater social surplus only from the variety effect. Both the market and the regulator value increased variety the same, since the first terms on the right-hand side of both (17) and (23) are the same. The second term on the right-hand side of (17) arises from the price effect in the market.

When m^2/F_A is relatively small, the second term in (17) is large and there is a significant divergence between the market and the social optimum: the parameter space for which inefficient adoption occurs is relatively large. Under such conditions relatively few software varieties will be provided by the market for either network and the price differential of software will be large. When the fixed costs of software development are significant and thus the number of software firms is small is precisely when we would expect that the software firms would be Bertrand competitors.[23]

A difference in fixed costs is a sufficient condition for the existence of parameter values where inefficient adoption occurs. However, it is also a necessary condition for the existence of parameter values where adoption is

[23]When the fixed costs of software development are small and there are many firms in the software industry, the assumption of monopolistic competition is appropriate. When the software market is monopolistically competitive, the technology chosen by the market is the same one a regulator would select in either regulatory scheme. See footnote 13, and our working paper, Church and Gandal (1991).

inefficient. This follows from observing the outcome in the market and the choice of a regulator if $F_A = F_B$. From (22) the regulator prefers the technology with the lowest hardware marginal cost. When the two technologies are priced at marginal cost then, from (12), the technology with the lowest marginal cost of hardware will have a greater number of software varieties and, using Lemma 1, a lower price of software. As a result, from (5a), the technology with the lower marginal cost of hardware will be adopted in the market and there is no divergence between the choice of a regulator and the market outcome.

8. Conclusion

In this paper we considered technologies with network externalities generated by complementary products. When the complementary products are incompatible between hardware technologies, market standardization depends not only on the difference in hardware costs, but also on software development costs. The role of the software development costs is in determining the number of software firms which can enter. The software development costs determine both the relative number of complementary products and the nature of the competition in the software industry. If the development costs of software are large vis-à-vis the consumer cohort, then the number of software products will be small and the appropriate competition in the software industry will be Bertrand. Under these circumstances the adoption pattern of the market is inefficient. The technology with the lower costs of software development is over-adopted by the market.

Can a regulator correct such an inefficiency? Our results indicate that when a regulator is limited to mandating adoption of a hardware technology, the regulator cannot improve on the market outcome. There is no role for regulation since the hardware choice of the regulator and the market outcome coincide. On the other hand, if price and entry regulation of the software industry is possible, we establish that a well informed regulator, i.e. one who has the relevant cost information prior to selecting the technology, can improve on the market outcome and that there is a role for regulation.

Appendix

A.1. Demand for software

We now derive the demand for software; by duality we can minimize expenditure subject to a fixed utility level. We minimize

$$E = \sum_{i=1}^{N_h} \rho_i^h x_i$$

subject to a fixed sub-utility level, z. The Lagrangian is

$$L=\sum_{i=1}^{N_h} \rho_i^h x_i + q_h\left[z-\left(\sum_{i=1}^{N_h} x_i^{1/\beta}\right)^{\beta}\right],$$

$$\frac{\partial L}{\partial x_i}=\rho_i^h - q_h\left(\sum_{i=1}^{N_h} x_i^{1/\beta}\right)^{\beta-1} x_i^{(1-\beta)/\beta}, \quad \forall i,$$

$$\frac{\partial L}{\partial q_h}=z-\left(\sum_{i=1}^{N_h} x_i^{1/\beta}\right)^{\beta}.$$

Setting $\partial L/\partial x_i=0$, substituting $z^{(\beta-1)/\beta}=(\sum_{i=1}^{N_h} x_i^{1/\beta})^{\beta-1}$ and solving for $x_i^{1/\beta}$ yields

$$x_i^{1/\beta}=\left(\frac{q_h}{\rho_i^h}\right)^{1/(\beta-1)} z^{1/\beta}, \quad \forall i. \tag{A.1}$$

Setting $\partial L/\partial q_h=0$, and substituting the above expression for $x_i^{1/\beta}$ yields

$$q_h=\left(\sum_{i=1}^{N_h} (\rho_i^h)^{-1/(\beta-1)}\right)^{1-\beta}.$$

This is the price index of software in the first-stage budgeting problem since it measures the cost of increasing consumption of z by one unit.

Solving (A.1) for x_i and substituting into the definition of software expenditure yields the expenditure function

$$\sum_{i=1}^{N_h} \rho_i^h x_i = q_h z = e(\rho, z).$$

Define I to be the disposable income spend on software, $(y-p_h)/2$. Using the identity that $I=e(\rho, z(\rho, I))$, then the indirect utility function $z(\rho, I)$ is

$$z=\frac{I}{q_h}.$$

Since

$$-\frac{\partial z}{\partial \rho_i^h}=\left(\frac{y-p_h}{2q_h^2}\right)\frac{\partial q_h}{\partial \rho_i^h}=\left(\frac{y-p_h}{2q_h^2}\right)\frac{(q_h)^{\beta/(\beta-1)}}{(\rho_i^h)^{\beta/(\beta-1)}} \quad \text{and} \quad \frac{\partial z}{\partial I}=\frac{1}{q_h},$$

then

$$x_i = \frac{-\partial z/\partial \rho_i^h}{\partial z/\partial I} = \frac{(y - p_h) q_h^{1/(\beta-1)}}{2(\rho_i^h)^{\beta/(\beta-1)}}, \quad \forall i,$$

by Roy's Identity.

A.2. Software price equilibrium

We now derive the equilibrium software price. Differentiating (7) with respect to ρ_i^h, we obtain the following first-order condition:

$$(q_h)^{1/(\beta-1)}(\rho_i^h)^{\beta/(1-\beta)} + \frac{\beta}{1-\beta}(\rho_i^h - s)(q_h)^{1/(\beta-1)}(\rho_i^h)^{(2\beta-1)/(1-\beta)}$$

$$+ \frac{1}{\beta-1}(\rho_i^h - s)(q_h)^{2/(\beta-1)}(\rho_i^h)^{2\beta/(1-\beta)} = 0. \qquad \text{(A.2)}$$

In a symmetric equilibrium, $\rho_i^h = \rho^h$, $\forall i$, and hence $q_h = \rho^h N_h^{-(\beta-1)}$. Making these substitutions into (A.2) we obtain

$$\rho^h + (\rho^h - s)\frac{\beta}{1-\beta} + \frac{\rho^h - s}{(\beta-1)N_h} = 0. \qquad \text{(A.3)}$$

Solving (A.3) for ρ^h gives (8).

A.3. Proof of Proposition 4

Assume $F_B > F_A$, then the social planner is indifferent between the two technologies if, from (22),

$$[y - c_A] = \left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)} [y - c_B]. \qquad \text{(A.4)}$$

If technology A is over-adopted, then when the social surplus provided by the two technologies is the same, the market surplus of A is greater than the surplus provided by technology B. Substituting (A.4) into (16),

$$\left(\frac{F_A}{F_B}\right)^{[\beta(1-\beta)]/(\beta+1)} \left(m[y - c_B]\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)} - 2F_A\right)$$

$$\times \left(m(\beta-1)[y - c_B]\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)} + 2F_A\right)^{\beta-1}$$

$$>(m[y-c_B]-2F_B](m(\beta-1)[y-c_B]+2F_B)^{\beta-1}. \quad (A.5)$$

Let

$$k_1=\left(\frac{F_A}{F_B}\right)^{[\beta(1-\beta)]/(\beta+1)},$$

$$k_2=\left(m(\beta-1)[y-c_B]\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}+2F_A\right)^{\beta-1},$$

and $k_3=(m(\beta-1)[y-c_B]+2F_B)^{\beta-1}$. Substituting into (A.5) and rearranging terms, technology A is over-adopted if

$$\left(k_1k_2\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}-k_3\right)m[y-c_B]>2\mathrm{F}_Ak_1k_2-2F_Bk_3. \quad (A.6)$$

If $F_B>F_A$, then

$$\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}>\frac{F_A}{F_B}, \quad \forall\beta>1$$

and thus

$$F_B\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}>F_A.$$

Therefore (A.6) will be true if

$$\left(k_1k_2\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}-k_3\right)m[y-c_B]>\left(k_1k_2\left(\frac{F_A}{F_B}\right)^{(\beta-1)/(\beta+1)}-k_3\right)2F_B$$

or

$$m[y-c_B]>2F_B,$$

which is always true by the restriction on the parameters which ensures that $N_B>1$. Q.E.D.

References

Church, Jeffrey and Neil Gandal, 1991, Complementary network externalities and technological adoption, Discussion paper no. 128 (Department of Economics, University of Calgary).
Church, Jeffrey and Roger Ware, 1989, Adverse selection, market structure and innovation, Mimeo. (University of Calgary).

Chou, Chien-fu and Oz Shy, 1990, Network effects without network externalities, International Journal of Industrial Organization 8, 259–270.

Deaton, Angus and John Muellbauer, 1980, Economics and consumer behaviour (Cambridge University Press, Cambridge).

DeBrock, Lawrence and Robert Masson, 1985, Sooner or later?, Inventive rivalry and welfare, International Journal of Industrial Organziation 3, 421–438.

Dixit, Avinash and Joseph Stiglitz, 1977, Monopolistic competition and optimum product diversity, American Economic Review 67, 297–308.

Farrell, Joseph and Nancy Gallini, 1988, Second sourcing as a commitment: Monopoly incentives to attract competition, Quarterly Journal of Economics 103, 673–694.

Farrell, Joseph and Garth Saloner, 1985, Standardization, compatibility and innovation, Rand Journal of Economics 16, 70–83.

Farrell, Joseph and Garth Saloner, 1986a, Standardization and variety, Economics Letters 20, 71–74.

Farrell, Joseph and Garth Saloner, 1986b, Installed base and compatibility: Innovation, production preannouncements and predation, American Economic Review 76, 940–955.

Katz, Michael and Carl Shapiro, 1985, Network externalities, competition and compatibility, American Economic Review 75, 424–440.

Katz, Michael and Carl Shapiro, 1986a, Technology adoption in the presence of network externalities, Journal of Political Economy 94, 822–841.

Katz, Michael and Carl Shapiro, 1986b, Product compatibility choice in a market with technological progress, Oxford Economic Papers 38, 146–165.

Spence, Michael, 1976, Production selection, fixed costs, and monopolistic competition, Review of Economic Studies 43, 217–235.